01622 Advanced Dynamical Systems: Applications in Science and Engineering

Week 8: Time delays

Tobias K. S. Ritschel,
Assoc. Prof. in Stochastic Adaptive Control

Department of Applied Mathematics and Computer Science,
Technical University of Denmark

Last updated on March 27, 2026

Delay differential equations

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{1}$$

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \qquad (1)$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad i = 1, \ldots, m \qquad (2)$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0, 1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \, \mathrm{d}s$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0,1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \,\mathrm{d}s = 1 + t \tag{4}$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0,1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \,\mathrm{d}s = 1 + t \tag{4}$$

Solution for $t \in [1,2]$

$$x(t) = x(1) + \int_1^t \overbrace{x(s-1)}^{=1+(s-1)} \,\mathrm{d}s$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0, 1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \, \mathrm{d}s = 1 + t \tag{4}$$

Solution for $t \in [1, 2]$

$$x(t) = x(1) + \int_1^t \overbrace{x(s-1)}^{=1+(s-1)} \, \mathrm{d}s = 2 + \frac{1}{2}(t^2 - 1^2)$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0, 1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \, \mathrm{d}s = 1 + t \tag{4}$$

Solution for $t \in [1, 2]$

$$x(t) = x(1) + \int_1^t \overbrace{x(s-1)}^{=1+(s-1)} \, \mathrm{d}s = 2 + \frac{1}{2}(t^2 - 1^2) = \frac{3}{2} + \frac{1}{2}t^2 \tag{5}$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$

$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0, 1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \, \mathrm{d}s = 1 + t \tag{4}$$

Solution for $t \in [1, 2]$

$$x(t) = x(1) + \int_1^t \overbrace{x(s-1)}^{=1+(s-1)} \, \mathrm{d}s = 2 + \frac{1}{2}(t^2 - 1^2) = \frac{3}{2} + \frac{1}{2}t^2 \tag{5}$$

Derivatives

$$\dot{x}(t) = \begin{cases} 0, & t \leq 0, \\ 1, & t \in [0, 1], \\ t, & t \in [1, 2], \end{cases}$$

How smooth is the solution? An example

Initial value problem with delay differential equations

$$x(t) = 1, \qquad t \leq 0, \tag{3a}$$
$$\dot{x}(t) = x(t-1), \qquad t > 0 \tag{3b}$$

Solution for $t \in [0, 1]$

$$x(t) = x(0) + \int_0^t \overbrace{x(s-1)}^{=1} \,\mathrm{d}s = 1 + t \tag{4}$$

Solution for $t \in [1, 2]$

$$x(t) = x(1) + \int_1^t \overbrace{x(s-1)}^{=1+(s-1)} \,\mathrm{d}s = 2 + \frac{1}{2}(t^2 - 1^2) = \frac{3}{2} + \frac{1}{2}t^2 \tag{5}$$

Derivatives

$$\dot{x}(t) = \begin{cases} 0, & t \leq 0, \\ 1, & t \in [0,1], \\ t, & t \in [1,2], \end{cases} \qquad \ddot{x}(t) = \begin{cases} 0, & t \leq 0, \\ 0, & t \in [0,1], \\ 1, & t \in [1,2] \end{cases} \tag{6}$$

Steady states

In steady state, $x(t) = x_s$ for all t

Steady states

In steady state, $x(t) = x_s$ for all t

Steady state equations

$$0 = f(x_s, x_s, \ldots, x_s, u_s, d_s, p) \tag{7}$$

Steady states

In steady state, $x(t) = x_s$ for all t

Steady state equations

$$0 = f(x_s, x_s, \ldots, x_s, u_s, d_s, p) \tag{7}$$

The steady state is the same as for ordinary differential equations in the form

$$\dot{x}(t) = f(x(t), x(t), \ldots, x(t), u(t), d(t), p) \tag{8}$$

Conclusion: Time delays do not change the steady state

Stability – Linear systems

For linear systems, e.g., in the form

$$\dot{x}(t) = A(p)x(t) + G(p)x(t-\tau) + B(p)u(t) + E(p)d(t) \qquad (9)$$

the stability is determined by A, G, and the time delay τ

Stability – Linear systems

For linear systems, e.g., in the form

$$\dot{x}(t) = A(p)x(t) + G(p)x(t-\tau) + B(p)u(t) + E(p)d(t) \tag{9}$$

the stability is determined by A, G, and the time delay τ

Characteristic equation

$$P(\lambda) = \det\left(A + Ge^{-\tau\lambda} - \lambda I\right) = 0 \tag{10}$$

In general, infinitely many solutions

Graphical stability analysis

Real and imaginary parts of characteristic function

$$P_r(\lambda) = \operatorname{Re} P(\lambda), \qquad P_i(\lambda) = \operatorname{Im} P(\lambda) \tag{11}$$

Graphical stability analysis

Real and imaginary parts of characteristic function

$$P_r(\lambda) = \operatorname{Re} P(\lambda), \qquad P_i(\lambda) = \operatorname{Im} P(\lambda) \tag{11}$$

Choose a grid of complex values

$$\lambda_{mn} = a_m + ib_n, \qquad m = 1, \ldots, M, \qquad n = 1, \ldots, N \tag{12}$$

Graphical stability analysis

Real and imaginary parts of characteristic function

$$P_r(\lambda) = \mathrm{Re}\, P(\lambda), \qquad P_i(\lambda) = \mathrm{Im}\, P(\lambda) \tag{11}$$

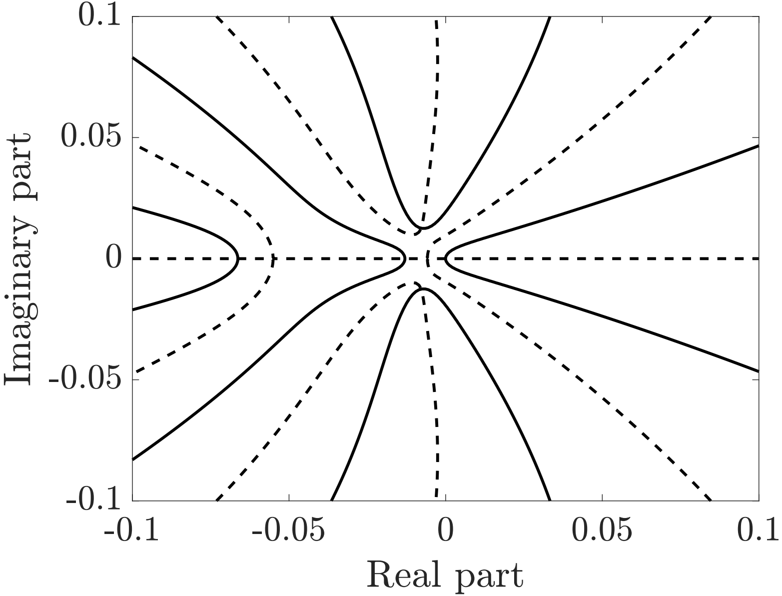
Choose a grid of complex values

$$\lambda_{mn} = a_m + ib_n, \quad m = 1, \ldots, M, \quad n = 1, \ldots, N \tag{12}$$

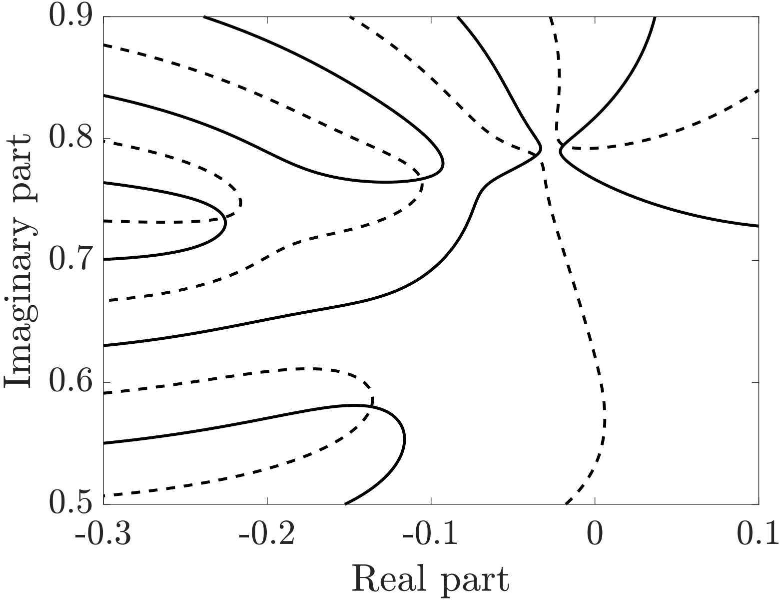
Plot the zero-contours of P_r and P_i

The intersection between the contours indicate the roots

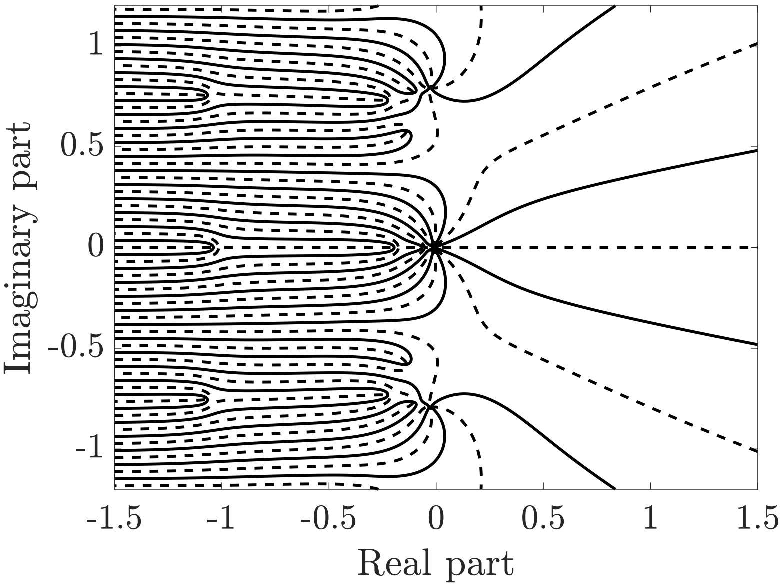
Graphical stability analysis



Graphical stability analysis



Graphical stability analysis



Numerical computation of the roots

Decision variables: a and b ($\lambda = a + ib$)

Numerical computation of the roots

Decision variables: a and b ($\lambda = a + ib$)

Algebraic equations

$$F(a,b) = P_r(a+ib) = 0, \quad (13a)$$
$$G(a,b) = P_i(a+ib) = 0 \quad (13b)$$

Numerical computation of the roots

Decision variables: a and b ($\lambda = a + ib$)

Algebraic equations

$$F(a, b) = P_r(a + ib) = 0, \tag{13a}$$

$$G(a, b) = P_i(a + ib) = 0 \tag{13b}$$

Two nonlinear equations in two variables that can be solved using, e.g., Matlab's fsolve

Initial guess: Use the graphical analysis

Stability – Nonlinear systems

For nonlinear systems in the general form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \qquad (14)$$

the stability is determined by A, G_i, and τ_i for $i = 1, \ldots, m$

Stability – Nonlinear systems

For nonlinear systems in the general form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{14}$$

the stability is determined by A, G_i, and τ_i for $i = 1, \ldots, m$

Characteristic equation

$$P(\lambda) = \det\left(A + \sum_{i=1}^{m} G_i e^{-\tau_i \lambda} - \lambda I\right) = 0 \tag{15}$$

Stability – Nonlinear systems

For nonlinear systems in the general form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{14}$$

the stability is determined by A, G_i, and τ_i for $i = 1, \ldots, m$

Characteristic equation

$$P(\lambda) = \det\left(A + \sum_{i=1}^{m} G_i e^{-\tau_i \lambda} - \lambda I\right) = 0 \tag{15}$$

Matrices

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix},$$

Stability – Nonlinear systems

For nonlinear systems in the general form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{14}$$

the stability is determined by A, G_i, and τ_i for $i = 1, \ldots, m$

Characteristic equation

$$P(\lambda) = \det\left(A + \sum_{i=1}^{m} G_i e^{-\tau_i \lambda} - \lambda I\right) = 0 \tag{15}$$

Matrices

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad G_i = \frac{\partial f}{\partial z_i} = \begin{bmatrix} \frac{\partial f_1}{\partial z_{i,1}} & \cdots & \frac{\partial f_1}{\partial z_{i,k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_{i,1}} & \cdots & \frac{\partial f_n}{\partial z_{i,k}} \end{bmatrix}, \tag{16}$$

$i = 1, \ldots, k$

Open-loop simulation

Numerical simulation

Programming language	Simulator	Note
Matlab	dde23	Constant time delays
Matlab	**ddesd**	General time delays
Matlab	ddensd	Neutral DDEs
Python	**JiTCDDE**[1]	General time delays

[1] `https://jitcdde.readthedocs.io/en`

Open-loop simulation

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p), \tag{17a}$$

$$z_i(t) = x(t - \tau_i), \quad i = 1, \ldots, m \tag{17b}$$

Open-loop simulation

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p), \qquad (17a)$$

$$z_i(t) = x(t - \tau_i), \quad i = 1, \ldots, m \qquad (17b)$$

Zero-order hold parametrization

$$u(t) = u_k, \qquad t \in [t_k, t_{k+1}[, \qquad (18a)$$

$$d(t) = d_k, \qquad t \in [t_k, t_{k+1}[\qquad (18b)$$

Open-loop simulation

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p), \tag{17a}$$
$$z_i(t) = x(t - \tau_i), \quad i = 1, \ldots, m \tag{17b}$$

Zero-order hold parametrization

$$u(t) = u_k, \qquad t \in [t_k, t_{k+1}[, \tag{18a}$$
$$d(t) = d_k, \qquad t \in [t_k, t_{k+1}[\tag{18b}$$

Open-loop simulation:

1. Create a function that, for given time t, returns u_k and d_k, and call dde23/ddesd/JiTCDDE <u>once</u> for all control intervals

Open-loop simulation

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p), \tag{17a}$$
$$z_i(t) = x(t - \tau_i), \quad i = 1, \ldots, m \tag{17b}$$

Zero-order hold parametrization

$$u(t) = u_k, \qquad t \in [t_k, t_{k+1}[, \tag{18a}$$
$$d(t) = d_k, \qquad t \in [t_k, t_{k+1}[\tag{18b}$$

Open-loop simulation:

1. Create a function that, for given time t, returns u_k and d_k, and call dde23/ddesd/JiTCDDE once for all control intervals
2. For each control interval, use the solution structure from the previous call to dde23/ddesd/JiTCDDE as the "history" input

Nuclear reactor models

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Mass balance equations (ρ_s is the salt density)

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{20a}$$

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Mass balance equations (ρ_s is the salt density)

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{20a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t))D \tag{20b}$$

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Mass balance equations (ρ_s is the salt density)

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{20a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t))D \tag{20b}$$

$$C_{i,in}(t) = e^{-\lambda_i \tau} C_i(t - \tau), \tag{20c}$$

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Mass balance equations (ρ_s is the salt density)

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{20a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t))D \tag{20b}$$

$$C_{i,in}(t) = e^{-\lambda_i \tau} C_i(t - \tau), \tag{20c}$$

$$D = \frac{F}{V}, \quad f = \rho_s F, \quad F = Av, \quad \tau = L/v \tag{20d}$$

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Mass balance equations (ρ_s is the salt density)

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{20a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t))D \tag{20b}$$

$$C_{i,in}(t) = e^{-\lambda_i \tau} C_i(t - \tau), \tag{20c}$$

$$D = \frac{F}{V}, \quad f = \rho_s F, \quad F = Av, \quad \tau = L/v \tag{20d}$$

Energy balance equations

$$\dot{T}_r(t) = \frac{f}{n_r}(T_{hx}(t - \tau/2) - T_r(t)) + \frac{Q_g(t)}{n_r c_P}, \tag{21a}$$

Nuclear reactor model 7 – Model 6 revisited

Reactivity and thermal reactivity

$$\rho(t) = \rho_{th}(t) + \rho_{ext}(t), \qquad \dot{\rho}_{th}(t) = -\kappa \dot{T}_r \tag{19}$$

Mass balance equations (ρ_s is the salt density)

$$\dot{C}_n(t) = \frac{\rho(t) - \beta}{\Lambda} C_n(t) + \sum_{i=1}^{m} \lambda_i C_i(t), \tag{20a}$$

$$\dot{C}_i(t) = \frac{\beta_i}{\Lambda} C_n(t) - \lambda_i C_i(t) + (C_{i,in}(t) - C_i(t))D \tag{20b}$$

$$C_{i,in}(t) = e^{-\lambda_i \tau} C_i(t - \tau), \tag{20c}$$

$$D = \frac{F}{V}, \quad f = \rho_s F, \quad F = Av, \quad \tau = L/v \tag{20d}$$

Energy balance equations

$$\dot{T}_r(t) = \frac{f}{n_r}(T_{hx}(t - \tau/2) - T_r(t)) + \frac{Q_g(t)}{n_r c_P}, \tag{21a}$$

$$\dot{T}_{hx}(t) = \frac{f}{n_{hx}}(T_r(t - \tau/2) - T_{hx}(t)) - \frac{k_{hx}}{n_{hx} c_P}(T_{hx}(t) - T_c) \tag{21b}$$

Nuclear reactor model 7 – time-varying time delays

If the velocity, v, is time-varying

$$D(t) = \frac{F(t)}{V}, \quad f(t) = \rho_s F(t), \quad F(t) = Av(t), \quad \tau(t) = L/v(t) \tag{22}$$

Time-varying time delays

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{23}$$

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{23}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad i = 1, \ldots, m \tag{24}$$

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{23}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad\qquad i = 1, \ldots, m \tag{24}$$

Time-varying time delays

$$\tau_i = \tau_i(t),$$

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{23}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad i = 1, \ldots, m \tag{24}$$

Time-varying time delays

$$\tau_i = \tau_i(t), \qquad \tau_i = \tau_i(u(t)), \tag{25a}$$

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{23}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad i = 1, \ldots, m \tag{24}$$

Time-varying time delays

$$\begin{aligned} \tau_i &= \tau_i(t), & \tau_i &= \tau_i(u(t)), \qquad (25a) \\ \tau_i &= \tau_i(x(t)), \end{aligned}$$

Delay differential equations

General form

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{23}$$

Memory states

$$z_i(t) = x(t - \tau_i), \qquad i = 1, \ldots, m \tag{24}$$

Time-varying time delays

$$\tau_i = \tau_i(t), \qquad \tau_i = \tau_i(u(t)), \tag{25a}$$

$$\tau_i = \tau_i(x(t)), \qquad \tau_i = \tau_i(t, x(t), u(t), d(t), p) \tag{25b}$$

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe
2. Picture a model of the "receiving" reactor with time delay, τ

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe
2. Picture a model of the "receiving" reactor with time delay, τ
   - The time delay is equal to length divided by velocity, $\tau = L/v$

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe
2. Picture a model of the "receiving" reactor with time delay, τ
    - The time delay is equal to length divided by velocity, $\tau = L/v$
3. Imagine that you reduce the velocity by a factor of 10

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe
2. Picture a model of the "receiving" reactor with time delay, τ
   - The time delay is equal to length divided by velocity, $\tau = L/v$
3. Imagine that you reduce the velocity by a factor of 10
4. What is the true "age" of the content in the pipe?

What are the underlying assumptions of time delays?

What do we assume about the process when we use time delays?

Thought experiment

1. Imagine two reactors that are connected by a pipe
2. Picture a model of the "receiving" reactor with time delay, τ
   - The time delay is equal to length divided by velocity, $\tau = L/v$
3. Imagine that you reduce the velocity by a factor of 10
4. What is the true "age" of the content in the pipe?
5. What is the age of the inlet stream in the receiving reactor?

Time-varying time delays – time-varying velocity

Instantaneous time delay (approximation)

$$\tau(t) = L/v(t) \tag{26}$$

Time-varying time delays – time-varying velocity

Instantaneous time delay (approximation)

$$\tau(t) = L/v(t) \tag{26}$$

Implicit equation for time delay (exact)

$$\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = L \tag{27}$$

Time-varying time delays – time-varying velocity

Instantaneous time delay (approximation)

$$\tau(t) = L/v(t) \tag{26}$$

Implicit equation for time delay (exact)

$$\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = L \tag{27}$$

Rewrite implicit equation (L is constant)

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = \dot{L} = 0 \tag{28a}$$

$$v(t) - (1 - \dot{\tau}(t))\,v(t - \tau(t)) = 0 \tag{28b}$$

Time-varying time delays – time-varying velocity

Instantaneous time delay (approximation)

$$\tau(t) = L/v(t) \tag{26}$$

Implicit equation for time delay (exact)

$$\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = L \tag{27}$$

Rewrite implicit equation (L is constant)

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = \dot{L} = 0 \tag{28a}$$

$$v(t) - (1-\dot{\tau}(t))\,v(t-\tau(t)) = 0 \tag{28b}$$

Differential equation for time delay (v is constant up to time t_0)

$$\dot{\tau}(t) = 1 - \frac{v(t)}{v(t-\tau(t))}, \qquad \tau(t_0) = L/v(t_0) \tag{29}$$

Be careful if v is discontinuous

Time-varying time delays – piecewise constant velocity

Piecewise constant velocity

$$v(t) = v_k, \qquad t \in [t_k, t_{k+1}[\tag{30}$$

Time-varying time delays – piecewise constant velocity

Piecewise constant velocity

$$v(t) = v_k, \qquad t \in [t_k, t_{k+1}[\tag{30}$$

Implicit equation for time delay

$$\begin{aligned}\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = (t - t_k)v_k + \sum_{j=1}^{K} v_{k-j}\Delta t \\ + (t_{k-K} - (t - \tau(t)))v_{k-K-1} = L, \quad t \in [t_k, t_{k+1}[\end{aligned} \tag{31}$$

Time-varying time delays – piecewise constant velocity

Piecewise constant velocity

$$v(t) = v_k, \qquad t \in [t_k, t_{k+1}[\tag{30}$$

Implicit equation for time delay

$$\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = (t - t_k)v_k + \sum_{j=1}^{K} v_{k-j}\Delta t + (t_{k-K} - (t - \tau(t)))v_{k-K-1} = L, \quad t \in [t_k, t_{k+1}[\tag{31}$$

Step 1: Determine K such that

$$(t - t_k)v_k + \sum_{j=1}^{K} v_{k-j}\Delta t < L \le (t - t_k)v_k + \sum_{j=1}^{K+1} v_{k-j}\Delta t \tag{32}$$

Time-varying time delays – piecewise constant velocity

Piecewise constant velocity

$$v(t) = v_k, \qquad\qquad t \in [t_k, t_{k+1}[\tag{30}$$

Implicit equation for time delay

$$\begin{aligned}\int_{t-\tau(t)}^{t} v(s)\,\mathrm{d}s = (t - t_k)v_k + \sum_{j=1}^{K} v_{k-j}\Delta t \\ + (t_{k-K} - (t - \tau(t)))v_{k-K-1} = L, \quad t \in [t_k, t_{k+1}[\end{aligned} \tag{31}$$

Step 1: Determine K such that

$$(t - t_k)v_k + \sum_{j=1}^{K} v_{k-j}\Delta t < L \leq (t - t_k)v_k + \sum_{j=1}^{K+1} v_{k-j}\Delta t \tag{32}$$

Step 2: Determine $\tau(t)$

$$\tau(t) = t - t_{k-K} + \frac{1}{v_{k-K-1}}\left(L - (t - t_k)v_k - \sum_{j=1}^{K} v_{k-j}\Delta t\right) \tag{33}$$

Time delays as partial differential equations

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{34}$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t, s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t, s),$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t,s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t,s), \qquad z_i(t,0) = x(t),$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t,s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t,s), \qquad z_i(t,0) = x(t), \qquad s \in [0,1], \tag{35}$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t,s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t,s), \qquad z_i(t,0) = x(t), \qquad s \in [0,1], \tag{35}$$

Method of lines (first-order upwinded finite difference scheme)

$$z_{i,0}(t) = x(t), \tag{36a}$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \dots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t, s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t, s), \quad z_i(t, 0) = x(t), \quad s \in [0, 1], \tag{35}$$

Method of lines (first-order upwinded finite difference scheme)

$$z_{i,0}(t) = x(t), \tag{36a}$$

$$\dot{z}_{i,n}(t) = -\frac{1}{\tau_i}\frac{z_{i,n}(t) - z_{i,n-1}(t)}{\Delta s}, \quad \Delta s = \frac{1}{N}, \quad n = 1, \dots, N, \tag{36b}$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \dots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t, s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t, s), \quad z_i(t, 0) = x(t), \quad s \in [0, 1], \tag{35}$$

Method of lines (first-order upwinded finite difference scheme)

$$z_{i,0}(t) = x(t), \tag{36a}$$

$$\dot{z}_{i,n}(t) = -\frac{1}{\tau_i}\frac{z_{i,n}(t) - z_{i,n-1}(t)}{\Delta s}, \quad \Delta s = \frac{1}{N}, \quad n = 1, \dots, N, \tag{36b}$$

$$z_i(t) = z_{i,N}(t) \tag{36c}$$

Time delays as partial differential equations

System

$$\dot{x}(t) = f(x(t), z_1(t), \ldots, z_m(t), u(t), d(t), p) \tag{34}$$

Transport equation

$$\frac{\partial z_i}{\partial t}(t, s) = -\frac{1}{\tau_i}\frac{\partial z_i}{\partial s}(t, s), \quad z_i(t, 0) = x(t), \quad s \in [0, 1], \tag{35}$$

Method of lines (first-order upwinded finite difference scheme)

$$z_{i,0}(t) = x(t), \tag{36a}$$

$$\dot{z}_{i,n}(t) = -\frac{1}{\tau_i}\frac{z_{i,n}(t) - z_{i,n-1}(t)}{\Delta s}, \quad \Delta s = \frac{1}{N}, \quad n = 1, \ldots, N, \tag{36b}$$

$$z_i(t) = z_{i,N}(t) \tag{36c}$$

The differential equations (34) and (36b) are ordinary

See [1] for more details and other ways to approximate time delays

Questions?

Bibliography I

[1] T. K. S. Ritschel, A. T. Reenberg, P. E. Carstensen, J. Bendsen, and J. B. Jørgensen, "Mathematical meal models for simulation of human metabolism." arXiv: 2307.16444, 2024. Preprint.

Bibliography II

[7] T. K. S. Ritschel and S. Stange, “Numerical optimal control for delay differential equations: A simultaneous approach based on linearization of the delayed state.” arXiv:2410.02687, 2024. Preprint.

[8] T. K. S. Ritschel, “Numerical optimal control for distributed delay differential equations: A simultaneous approach based on linearization of the delayed variables.” arXiv:2410.15083, 2024. Preprint.